IX: A Protected Dataplane Operating System for High Throughput and Low Latency

Belay, A. et al.

Proc. of the 11th USENIX Symp. on OSDI, pp. 49-65, 2014.

Reviewed by Chun-Yu and Xinghao Li

Summary

In this paper, the authors introduce a new approach to improve the throughput and reduce the latency with a customized dataplane operating system, called IX. The motivation is based on the fact that the operating system is a major source of the performance bottleneck as described in the paper. First of all, the underlying logic of modern server operating system was developed according to the hardware assumptions that mismatch those used in some specific applications in datacenters. In addition, the modern server operating systems such as Linux family are optimized for multi threaded processings. In another word, they are developed for general purpose tasks. However, this optimization does not work at best under specific conditions, such as servers that handles a large amount of small packets, which is very common in data centers. As a result, it is hard to both increase the throughput and reduce the latency at the same time.

IX breaks the tradeoff between those two factors with some innovative improvements and also takes resource and security into consideration. The key concept of IX is to separate the dataplane apart from the control plane and allocate dedicated hardware resources and network processing queues to dataplane procedures to avoid excessive synchronization overheads. Besides, IX also has some features that will further advance its networking performance. For instance, it applies the zero-copy feature in order to reduce the number of copy operations in memory, which will consume a large amount of CPU time. It also modifies the batching policy by limiting its usage upon congestion and bounding the batched packets to a maximum amount, which also leads to higher throughput and lower latency. With such designs and implementations, IX beats original Linux kernel and m-TCP in all of the evaluations described in the paper as it has the significantly better performance than others under same workload and hardware specifications. Also, the performance throttling point for IX is very close to the physical limits.

Highlights

Other than the key contribution mentioned in the last section, the first thing we like about this paper is that the authors directly pointed out the challenges for datacenter applications and stated why current operating systems do not solve the real problem in data center. There are issues of 1) microsecond tail latency, 2) high packet rates, 3) security (protection), and 4) resource efficiency. Current solutions only address some of the aspects, but not all of them.

For microsecond latency, the authors pointed out that each user request can involve up to hundreds of servers. Therefore, long tail latency of RPC requests across data center must be taken into design consideration.

For high packet rates, the authors pointed out that the sizes of the requests are often small. Therefore, servers are dealing with large number of small requests concurrently. They state that memcached servers at Facebook do not use TCP and use UDP because of this issue.

For security and protection, they propose that isolation between applications is required. As for resource utilization, the system should use fewest resource to deal with the issues we mentioned earlier in order to use the rest of the resource for other applications or save power.

The author also pointed out a very important aspect which is the basic assumption mismatch between hardware and OSes. Nowadays, operating systems trade off both latency and throughput for fine-grain scheduling. Therefore, they can not meet the requirement of high packet rate and latency.

The next thing we like is that they adopted the “run each packet to complete” concept from the design of middlebox and applied it to the operating system. By doing so, the operating system is able scale on many cores because the network flows can now be distributed to different queues of different cores. Another good reason to adopt run to completion is that the system is now able to use polling to avoid interrupt overhead between network stack and application logic.

Also, the authors adopted adaptive batching in every stage of network stack to increase packet rate, since running batch is able to decrease the system call transition overhead. The adaptive batch has two features: 1) never wait for requests and 2) there is an upper bound on the number of batches. The first one can ensure that the requests will not be hold off while waiting for other requests and can improve efficiency during congestion. The second feature makes sure that cache capacity will not be exceeded.

The next thing we like is the design of zero-copy API. Because there is no copying in the API, packet rate and latency is improved. Instead, application need to inform dataplane whenever a packet is finished processing.

The next thing worth mentioned is that each IX dataplane supports a single application and has its own NIC hardware queue. Furthermore, the control plane only allocates resource in coarse-grain manner and left the real fine-grain allocation according to real-time properties and cpusets. Therefore, overheads of time multiplexing can be avoided.

Hierarchical timing wheel implementation is a smart implementation for managing network timeouts since the timeout for the network has maximum timeout value. By adopting this implementation, resource consumption can be reduced.

Improvement and Extension

Since improve resource efficiency is one of the goal for IX, the first thing this work can extended is to design a good resource allocation as the authors have mentioned in in Section IV. There are many aspects that need to be considered for resource allocation, including performance, fairness, energy consumption, and balance between system and application. For data centers, energy consumption may be one of the most important aspect. In order to reduce energy consumption, the OS may design to allocate resource to fewer amount of cores and let other cores have the chance to go to sleep.

As mentioned in Section 6, the design of IX has not yet been verified to specially support certain network protocols. Since in datacenter everything is under control, IX can be improved by optimizing the design to the network protocol the owner intend to use. Therefore, further improvement can be achieved. Also, it can be further improved by looking into different commonly used applications.

In the evaluation part, the authors only tested the case when there is only one server. In data center, it is more likely that multiple servers need to communicate with each other in order to service a single request. Some more detailed experiment for multiple server condition will be helpful to verify whether IX is able to support this kind of condition. Also, because the data the authors use to conduct the experiment is limited to benchmark and memcached service in the paper, it may be helpful to show that whether IX has better performance than other design regarding to different kinds of services that have traffic patterns. At production level, different services mix up, and the proportion of each individual service may change over time. So, benchmarks cannot represent the situation in production environments as they only contain individual tests or mixed evaluations under a fixed proportion. Therefore, it is necessary to perform a series of evaluations in real production environments to provide more persuasive results.

In the latency and throughput evaluation (fig. 5), the results are given by the 99th percentile and average. It will be more convincing to be also presented by different levels of percentile for latency vs throughput. Since 99th percentile can be considered as including some of the worst case scenario, it is likely that this kind of situation rarely occurs. An average measurement is really useful to answer this question, but it is very useful to have a more general idea of how other percentile performs, such as 50th percentile that indicating the behavior of half of the time.

Since there are four system challenges mentioned in Section 2, this paper seems to omit the evaluation of the last two parts, i.e. resource and security. For resource, we are curious that what are CPU and RAM utilization that IX compared to others. It will be even better if there are results for power consumption for the server. Security is also a crucial factor for server operating systems, especially in datacenter since a single vulnerability may cause the entire network to crash, which is a disaster for commercial applications. Thus, it is necessary to perform certain security and vulnerability tests before deployment.

Finally, this paper only provides the advantages of IX without state its limitations such as the kinds of applications it does not work well. In my opinion, when discussing a product, it is better to be more objective to state both its strengths and weaknesses rather than completely avoid discussing the drawbacks. In addition, there is no information about whether IX has been adapted by some production applications or works only for test in the lab. A high adoption rate means it is robust and well accepted by the public, which is also a strong support for the strengths of IX.